شرح وتقديم
م. حاسن الحارثي

بسم الله الرحمن الرحيم

شرح تفصيلي حول عاملي الزيادة أو النقصان السابق واللاحق في لغة السي شارب

هذا الشرح مقدم لأعضاء مجموعة C# Arab في تطبيق Telegram
أسأله سبحانه أن يجعله خالصاً لوجهه الكريم

حديثنا خصيصاً عن : مفهوم معامل الزيادة والنقصان (- - , ++).
يقصد بعامل الزيادة هو زيادة للقيمة المسندة للمتغير بواحد وكذلك الحال بالنسبة لمعامل النقصان هو إنقاص للقيمة المسندة للمتغير بواحد.

نبدأ بشكل عملي من هنا :
لنفترض أن لدينا المثال التالي :

```
int x = 1;
Console.WriteLine(x);
```

لو قمنا بتنفيذ الكود سنحصل على النتيجة التالية:



حتى الآن، عملنا يسير بشكل صحيح وهو أن القيمة المسندة للمتغير نتيجتها بعد التنفيذ = 1 .
دعونا الآن نقوم بزيادة هذه القيمة بواحد، بمعنى نضيف على القيمة السابقة قيمة أخرى = 1 ولكن بدون عامل الزيادة أو النقصان، كيف يكون ذلك؟
شاهد المثال التالي والذي قمنا فيه بكتابة نفس الكود السابق ولكن نود إضافة قيمة جديدة للمتغير(بمعنى نضيف على القيمة القديمة قيمة جديدة =1 فقط) :

```
int x = 1;
x = x + 1;
Console.WriteLine(x);
```

المفترض أن تكون نتيجة الكود = 2 كما في الصورة التالية :



لماذا النتيجة = 2 ؟
الشرح : فقط قمنا بزيادة بمقدار 1 للقيمة القديمة المسندة للمتغير
القيمة القديمة :

```
int x = 1;
```

القيمة الجديدة :

```
x = x + 1;
```

ناتج العملية = 2

يفترض حتى الآن الأمور واضحة وجلية.

شرح وتقديم
م. حاسن الحارثي

دعونا نأخذ مثال آخر على الزيادة ولكن أيضاً بدون استخدام عاملي الزيادة والنقصان (++,--)!

شاهد المثال التالي الذي يقوم بزيادة قيمة المتغير بدون استخدام عاملي الزيادة والنقصان :

```
int x = 1;
x += 1;
Console.WriteLine(x);
```

نتيجة الكود = 2



ان شاء الله يكون الأمر واضح حتى الآن!
لنعود قليلاً لمثالنا قبل الحالي اللي هو :

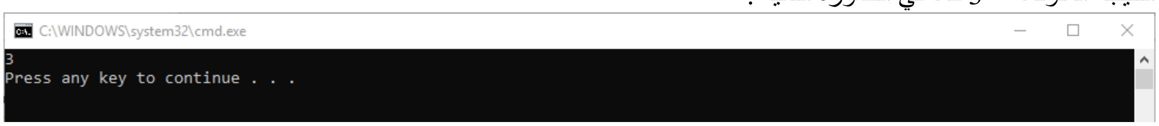
```
int x = 1;
x = x + 1;
Console.WriteLine(x);
```

اتفقنا أن نتيجته ستكون = 2 ، وشرحنا لماذا حصلنا على النتيجة 2 !

لنقم الآن بإضافة هذا المثال لمثالنا الحالي وستكون الصورة الكاملة بهذه الطريقة :

```
int x = 1;
x = x + 1;
x += 1;
Console.WriteLine(x);
```

النتيجة المتوقعة = 3 كما في الصورة التالية :



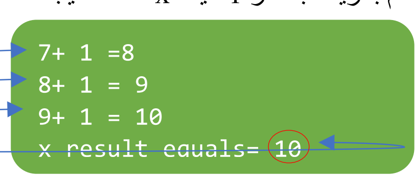
حسناً، السؤال – لماذا حصلنا على الرقم 3؟
الجواب : قمنا فقط بإسناد قيمة جديدة للمتغير بمقدار 1 وحاصل الجمع لكل القيم المسندة للمتغير هو = 3 .
كل ما سبق هو مقدمة وليس موضوعنا أصلا ولكن للتبسيط وجب ذكره.

لنأخذ المثال التالي لتقريب المفهوم أكثر :
لدينا الكود التالي :

```
int x = 7;
x = x + 1;
x += 1;
x++;
Console.WriteLine(x);
```

أهاا! في الكود الحالي يوجد معامل الزيادة وقام بأخذ قيمة الـ x ، ترى كم ستكون النتيجة؟!
أفترض أنك توقعت النتيجة التالية وهي = 10 .
لماذا 10؟
الجواب : معامل الزيادة في مثالنا هذا قام بالزيادة بمقدار 1 لقيمة x لذلك النتيجة كتفصيل حصلت كالتالي :

```
int x = 7;
x = x +1;
x += 1;
x++;
Console.WriteLine (x);
```



تمام!

الآن لو قمنا بكتابة الكود التالي :

```
int x = 7;
x++;
++x;
Console.WriteLine(x);
```

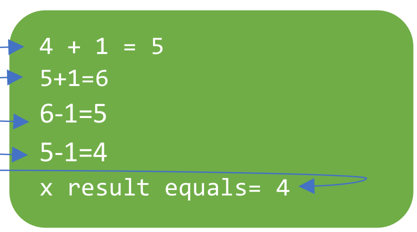
في مثالنا هذا استخدمنا معامل الزيادة قبل المتغير وبعده، حسناً- برأيك، ماهي النتيجة المتوقعة لناتج هذا الكود؟
إذا كان جوابك ذهنياً (في عقلك) = 9 فأنت تفكر بطريقة جيدة لأنه بالفعل نتيجة الكود = 9. إن خالف جوابك الرقم 9 ارجع واقرأ من أول كلمة في الملف وبعمق لتصل لماذا حصلنا على النتيجة = 9.

نتيجة الكود :



حسناً، لماذا 9؟
الجواب :
قيمة الـ x هي 7 متفقين! – حسناً، قمنا بوضع معامل الزيادة بمقدار 1 (x++) ولكن لو نلاحظ الآن أن معامل الزيادة أتى بعد اسم المتغير وليس قبله- لا يهم – كل ما يهمنا هو أن النتيجة ستكون = 8 لأن مقدار الزيادة أصبح يساويا واحداً
أما بالنسبة لمعامل الزيادة بمقدار واحد (++x) ونلاحظ أن المعامل أتى قبل اسم المتغير فنتيجته = 1 كما في معامل الزيادة الذي أتى بعد اسم المتغير!!
دعونا نأخذ مثال على معامل النقصان (--) ونوضح ذلك بمثال كما في الكود التالي :
الكود :

```
int x = 4;
x++;
++x;
x--;
--x;
Console.WriteLine(x);
```



هاه! ما النتيجة المتوقعة؟
ستكون نتيجة الكود = 4!
لماذا 4؟ يفترض بالفعل أن تكون عارفاً بكيفية حصولنا على النتيجة 4!.
نتيجة الكود :



حتى الآن سنصل إلى :
معنى ذلك أو المفهوم أن كلا عاملي الزيادة أو النقصان بالمقدار المحدد يؤديان نفس المهمة أو الغرض؟؟
الجواب: نعم! تماماً، يؤدون نفس المهمة!!.
ولكن هل هناك فرق؟ وما الفائدة؟
الجواب : نعم هناك فرق!.

نوضح ذلك:
دعونا نأخذ معامل الزيادة بعد اسم المتغير :

```
x++;
```

هذا يسمى : Post Fix - بعد

ماذا يعني؟

حسناً : معنى ذلك أن معامل الزيادة (++) أتى بعد المتغير (ركزوا في الثلاث كلمات التي عليها خط)

وهنا – دعونا نأخذ معامل الزيادة بعد المتغير :

هذا يسمى : Pre Fix - قبل

```
++x;
```

ماذا يعني؟

حسناً : معنى ذلك أن معامل الزيادة (++) أتى قبل المتغير (ركزوا في الثلاث كلمات التي عليها خط)

ننتقل لمثال جديد.

لدينا الكود التالي :

```
int x = 10;
int a = x++;
Console.WriteLine(a);
```

ماذا تتوقع نتيجة الكود؟

الجواب :



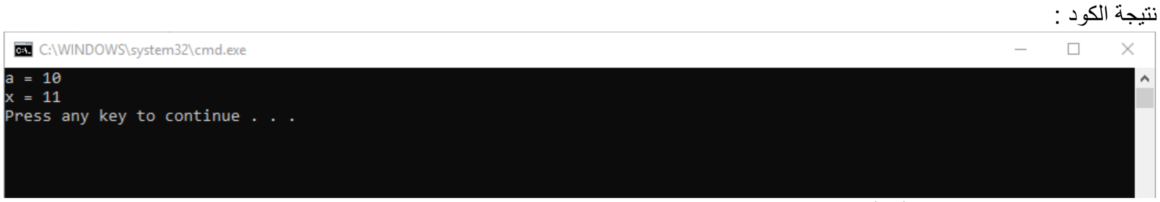
الجواب = 10!!!!!! لماذا؟ نحن بالفعل أضفنا معامل الزيادة بمقدار 1 لقيمة المتغير x من المفترض أن تكون قيمة x الجديدة =11 ! ما الذي حصل؟

لاتقلق! معامل الزيادة فعلاً قادم بدوره وقام بعمل الزيادة بمقدار 1

لاحظ ودقق معي في الكود:

```
int x = 10;
int a = x++;
Console.WriteLine("a = "+ a);
Console.WriteLine("x = " + x);
```

نتيجة الكود :



بالفعل قيمة x زادت بمقدار 1، أي أصبحت = 11!

السؤال الآن، لماذا لم يأخذ المتغير a قيمته بعد الزيادة؟؟

الجواب: ببساطة لأن مقدار الزيادة في المتغير a هو Post Fix -بعد (ارجع للتعريف في الأعلى)

الذي نود الوصول إليه: ما المقصود بـ post fix هنا؟

الجواب: اذا عندنا مثلا المتغير a وأتى بعده معامل التخصيص (=) وأتى بعد معامل التخصيص (الذي هو علامة =) المتغير كـ post fix يعني معامل الزيادة أتى بعد اسم المتغير سيتم تنفيذ نتيجة معامل التخصيص قبل تنفيذ معامل الزيادة والتي تساوي في مثالنا = 10 . ولا يقوم بعمل مقدار الزيادة الذي يساوي واحداً

ما فهمت؟

طيب نشرح بالتفصيل الممل:

في مثالنا قيمة x = 10 تمام؟

تمام! وقيمة a = قيمة x + 1 (a= x++) تمام حتى الآن؟ يعني نضيف لها 1 لتصبح 11

تمام!

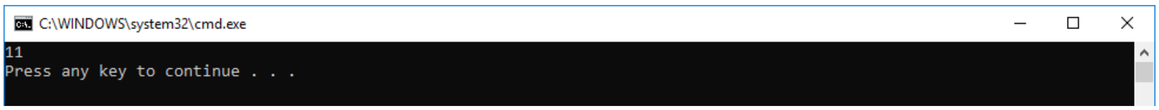
في مثالنا مقدار الزيادة أتى كـ post fix بمعنى (أنه أتى بعد المتغير x++) وعندما يأتي معامل الزيادة بعد اسم المتغير ينفذ أولاً معامل التخصيص الذي هو علامة = وفي مثالنا قيمة x تساوي 10 لذلك البرنامج نفذ أولا نتيجة معامل التخصيص وطبع 10 لأن مقدار الزيادة أتى كـ

وعندما يأتي معامل الزيادة كـ post fix يتم أولاً تنفيذ قيمة معامل التخصيص ولم يقم بعمل زيادة بمقدار 1 (بمعنى آخر أنه يأخذ القيمة القديمة ولا يلتفت للزيادة الحاصلة)
أتوقع بشكل كبير أن ذلك مفهوماً

طيب لو لدينا الكود التالي :

```
int x = 10;
int a = ++x;
Console.WriteLine(a);
```

أعتقد أنك فهمت المغزى الآن! وإن كانت إجابتك ذهنيا = 11 فهي صحيحة!
لماذا إذا تحقق هنا مقدار الزيادة؟!
أولا جواب الكود:



الشرح بالتفصيل : (لماذا تحقق مقدار الزيادة هنا؟)
الجواب:
اذا عندنا مثلا المتغير a ولدينا معامل التخصيص (=) وأتى بعد معامل التخصيص (الذي هو علامة =) معامل الزيادة قبل المتغير أي كـ pre fix سيتم تنفيذ نتيجة معامل الزيادة بالمقدار المحدد قبل تنفيذ نتيجة معامل التخصيص (أي ينفذ مقدار الزيادة الحاصلة للمتغير قبل القيمة القديمة للمتغير)
ونقيس ذلك أيضا على معامل النقصان (--) للأمثلة السابقة كما هي.

أرجو أن أكون وفقت في إيصال المعلومة وإزالة اللبس حول الموضوع

مع اطيب تحية ،،،،،،،،،،،،،،،